Program Database Toolkit

Many tasks in an integrated programming environment require access to program information for their implementations. The Program Database Toolkit (PDT) is a framework for analyzing source code written in several programming languages and for making rich program knowledge accessible to developers of static and dynamic analysis tools. PDT implements a standard program representation, the program database (PDB), that can be accessed in a uniform way through a class library supporting common PDB operations. Software tools can use this library to accomplish tasks such as

- documentation of program components;
- creation of graphic program browsers;
- insertion of performance instrumentation; and
- interface generation to call routines or for interlanguage bindings.

Figure 1 shows the PDT architecture and the tools that have been developed to use it.

PDT Intermediate Language Analysis

PDT utilizes state-of-the-art front-end parsers from the Edison Design Group and Mutek. Currently, PDT supports the C, C++, FORTRAN 77, and Fortran 90 languages. Each language front end produces its results in an "intermediate language" (IL) tree form. Although the IL program trees are similar, they are distinguished by the language constructs. Our IL Analyzers input the IL, walk the IL tree, and filter and reorganize the information about the parsed program into a more structured, standard format, the PDB.

The level of conversion provided by PDT is determined by the amount and detail of program information required by tools. Currently, the PDB contains all information on functions and classes, including template instantiations, and also
Acknowledgments:
The PDT project is a joint collaboration between the University of Oregon, the Advanced Computing Laboratory at Los Alamos National Laboratory, and Research Centre Jeulich in Germany. The IL Analyzers were written by Kathleen Lindlan and Robert Ansell Bell at the University of Oregon. DUCTAPE was developed by Bernd Mohr at the Research Centre Jeulich. SILOON and CHASM were developed by Craig Rasmussen at LANL.

Contact Information:
More information about PDT
www.acl.lanl.gov/pdtoolkit

Get PDT and other Advanced Computer Laboratory Software:
www.acl.lan.gov/software/

More information about IL Analyzers
Robert Ansell Bell, University of Oregon: bertie@cs.uoregon.edu

More information about DUCTAPE:
Bernd Mohr, Research Centre Jeulich BMohr@fz-jeulich.de

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

November 2001
LALP-01-255

A US DEPARTMENT OF ENERGY LABORATORY

contains information on templates and macros. The routine section lists source identification, parent class and access, signature, characteristics, functions called in each routine, and routine arguments. The class section specifies source information, characteristics, direct parent classes, member function IDs, and information on other members. The template and macro sections report source information, type, and text of each entity.

PDB Query and Management
A tool called DUCTAPE (C++ program Database Utilities and Conversion Tools APplication Environment) implements a C++ library that enables applications to access PDB files. With DUCTAPE, tools can read, merge, write, and, most importantly, query the PDB for specific program information. The structured form of the PDB is reflected in the DUCTAPE application programming interface, allowing easy access to all high-level source data, such as functions, classes, templates, source files, namespaces, and macros.

PDT Application
PDT is able to handle large source codes for DOE projects including the POOMA (Parallel Object-Oriented Methods and Applications) framework, the UPS (Unified Parallel System) framework, SAMRAI++, Overture, SAGE (SAIC Adaptive Grid Eulerian hydrocode), and POP (Parallel Ocean Program). It has primarily been applied for source-to-source program translation. The TAU (Tuning and Analysis Utilities) performance system uses PDT to instrument C, C++, FORTRAN 77, and FORTRAN 90 programs for profiling and tracing. SILOON (Scripting Interface Languages for Object-Oriented Numerics) has used PDT to automate the generation of "glue" code that enables C++ library routines to be called remotely from routines written in scripting languages. The PDT tool ensures that interfaces are used correctly and that interlanguage data representation differences are properly accounted for. Recently, this capability has been extended to the CHASM project to support automatic code generation for improved Fortran 90 and C++ interoperability.

PDT Availability
Version 2.0 of PDT is available for Web download. The release includes the IL Analyzers as well as DUCTAPE’s library and source code analysis tools. The TAU instrumentors are also available.